LEXIS·NEXIS
Xchange

Sources Clients Options History ECLIPSE™ Sign-Off Help

Search
Get a Document
Check a Citation

Source: Utility Patents
Type: ● Boolean ○ FREESTYLE ™

cache database | New Search |

prev ◄ Docs **1 - 10** of 16 ►next

| Cite List | KWIC™ | Full Text |

1. 5,892,939, Apr. 6, 1999, Emulator for visual display object files and method of operation thereof, Call, William L., Phoenix, Arizona Clawson, Laurence A., Cave Creek, Arizona Connolly, Paul S., Glendale, Arizona Freimark, Ronald J., Scottsdale, Arizona Gustin, Jay W., Scottsdale, Arizona Hodge, Michael L., Cave Creek, Arizona McGaugh, Paul, San Antonio, Texas Moore, Donald W., Phoenix, Arizona Rachlin, Elliott H., Scottsdale, Arizona Ramsdell, Steven C., Phoenix, Arizona, Honeywell Inc., Minneapolis, Minnesota (02)

   CORE TERMS: display, registered, non-native, emulator, window, visual, computer, storage, process control system, native...

2. 5,870,742, Feb. 9, 1999, Method and apparatus for enabling persistent attributes in an object oriented environment, Chang, David Yu, Austin, Texas High, Jr., Robert Howard, Round Rock, Texas Newcombe, Russell Lev, Round Rock, Texas, International Business Machines Corporation, Armonk, New York (02)

   CORE TERMS: persistent, storage, memory, persistence, interface, distributed, load-attribute, object-oriented, cache, data processing...

3. 5,864,864, Jan. 26, 1999, Method and apparatus for providing transparent persistent data support to foreign data types, Lerner, Benjamin, Palo Alto, California, Sun Microsystems, Inc., Mountain View, California (02)

   CORE TERMS: persistent, database, computer, transient, format, externalized, oriented, internalized, variable, storage...

4. 5,835,908, Nov. 10, 1998, Processing multiple database transactions in the same process to reduce process overhead and redundant retrieval from database servers, Bennett, John G., Bellevue, Washington Dalal, Ketan, Seattle, Washington, Microsoft Corporation, Redmond, Washington (02)

   CORE TERMS: database, cache, database-accessing, stream, lock, retrieved, computer system, identifier, server, caches...

5. 5,832,521, Nov. 3, 1998, Method and apparatus for performing consistent reads in multiple-server environments, Klots, Boris, Palo Alto, California Bamford, Roger J., Woodside, California, Oracle Corporation, Redwood Shores, California (02)

   CORE TERMS: derivation, database, node, server, storage, cache, layer, reside, executing, remote...

6. 5,815,195, Sep. 29, 1998, Subscriber information maintenance system and methods, Tam, Simon, Issaquah, Washington, Microsoft Corporation, Redmond, Washington (02)

   CORE TERMS: subscriber, database, interactive, interface, billing, specific information,

video, enhanced, modify, headend...

7. 5,799,210, Aug. 25, 1998, Method for allocating either private or shared buffer memory for storing data from sort operations in accordance with an assigned value or threshold value, Cohen, Jeffrey I., Mountain View, California Waddington, William H., Foster City, California, Oracle Corporation, Redwood Shores, California (02)

   CORE TERMS: buffer, memory, database, parameter, allocated, allocating, user, cache, tuning, disk...

8. 5,790,664, Aug. 4, 1998, Automated system for management of licensed software, Coley, Christopher D., Morgan Hill, California Wesinger, Jr., Ralph E., Livermore, California, Network Engineering Software, Inc., San Jose, California (02)

   CORE TERMS: licensing, computer, server, license, network, message, user, hierarchy, provider, database...
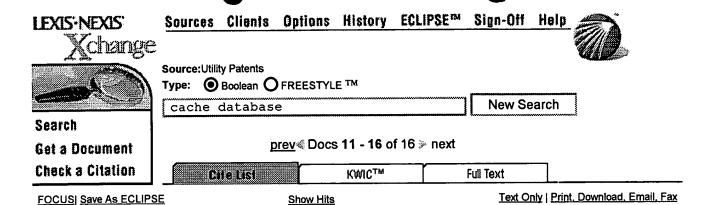
9. 5,761,660, Jun. 2, 1998, Computer program product and program storage device for database access using a shared electronic store in a multi-system environment having shared disks, Josten, Jeffrey William, Morgan Hill, California Masatani, Tina Louise, San Jose, California Mohan, Chandrasekaran, San Jose, California Narang, Inderpal S., Saratoga, California Teng, James Zu-Chia, San Jose, California, International Business Machines Corporation, Armonk, New York (02)

   CORE TERMS: database, designated, storage, updating, cached, update, buffer, log, store-dependent, lock...

10. 5,740,370, Apr. 14, 1998, System for opening cache file associated with designated file of file server only if the file is not subject to being modified by different program, Battersby, Clinton Brenton, Stoughton, Massachusetts Luth, James Gerard, Mansfield, Massachusetts, Battersby; Clinton, Stoughton, Massachusetts (04)

   CORE TERMS: cache, server, network, designated, opened, caching, computer, storage unit, backslash, user...

FOCUS| Save As ECLIPSE   Show Hits   Text Only | Print, Download, Email, Fax

| Cite List | KWIC™ | Full Text |

prev ◄ Docs **1 - 10** of **16** ►next

Source:	Utility Patents
Search Terms:	cache database
View:	CITE
Date/Time:	Saturday, June 19, 1999 - 8:33 PM EDT

Search | Get a Document | Check a Citation
Sources | Clients | Options | History | ECLIPSE | Feedback | Sign-Off | Help
About LEXIS-NEXIS | Terms and Conditions

LEXIS·NEXIS
Xchange

Source: Utility Patents
Type: ◉ Boolean ○ FREESTYLE ™

| cache database | | New Search |

Search

Get a Document

Check a Citation

| Cite List | KWIC™ | Full Text |

FOCUS| Save As ECLIPSE   Show Hits   Text Only | Print, Download, Email, Fax

11. 5,682,527, Oct. 28, 1997, Method and apparatus for block-level auditing and database recovery in a transaction processing system, Cooper, Thomas P., New Brighton, Minnesota Hill, Michael J., Vadnais Heights, Minnesota Konrad, Dennis R., Welch, Minnesota Nowatzki, Thomas L., Shoreview, Minnesota, Unisys Corporation, Blue Bell, Pennsylvania (02)

   CORE TERMS: audit, database, cache, outboard, processing, data processing, segment, disk, storage, packet...

12. 5,557,792, Sep. 17, 1996, Efficient data base access using a shared electronic store in a multi-system environment with shared disks, Josten, Jeffrey W., Morgan Hill, California Masatani, Tina L., San Jose, California Mohan, Chandrasekaran, San Jose, California Narang, Inderpal S., Saratoga, California Teng, James Z., San Jose, California, International Business Machines Corporation, Armonk, New York (02)

   CORE TERMS: database, cached, buffer, store-dependent, designated, updating, update, log, storage, lock...

13. 5,550,971, Aug. 27, 1996, Method and system for generating a user interface adaptable to various database management systems, Brunner, Hans, Denver, Colorado McCandless, Timothy P., Boulder, Colorado Sparks, Randall B., Louisville, Colorado Cuthbertson, Robert J., Boulder, Colorado Durand, Jacques, Louisville, Colorado Fogel, Steven M., Denver, Colorado, U S West Technologies, Inc., Boulder, Colorado (02), Date Transaction Recorded: May 29, 1998 ASSIGNMENT OF ASSIGNOR'S INTEREST (SEE DOCUMENT FOR DETAILS). U S WEST, INC. SUITE 480, 7800 E. ORCHARD ROAD LAW DEPARTMENT, INTELLECTUAL PROPERTY GROUP ENGLEWOOD, COLORADO 80111 Reel & Frame Number: 9197/0311Date Transaction Recorded: Jul. 07, 1998 ASSIGNMENT OF ASSIGNOR'S INTEREST (SEE DOCUMENT FOR DETAILS). U S WEST, INC. 1801 CALIFORNIA STREET DENVER, COLORADO 80202 MEDIAONE GROUP, INC. 188 INVERNESS DRIVE WEST ENGLEWOOD, COLORADO 80112 Reel & Frame Number: 9297/0308Date Transaction Recorded: Jul. 07, 1998 CHANGE OF NAME (SEE DOCUMENT FOR DETAILS). MEDIAONE GROUP, INC. 188 INVERNESS DRIVE WEST ENGLEWOOD, COLORADO 80112 Reel & Frame Number: 9297/0442

   CORE TERMS: database, dot, user, interface, fot, query, entity, layer, meta-model, char...

14. 5,455,942, Oct. 3, 1995, Partial page write detection for a shared cache using a bit pattern written at the beginning and end of each page, Mohan, Chandrasekaran, San Jose, California Narang, Inderpal S., Saratoga, California Teng, James Zu-chia, San Jose, California, International Business Machines Corporation, Armonk, New York (02)
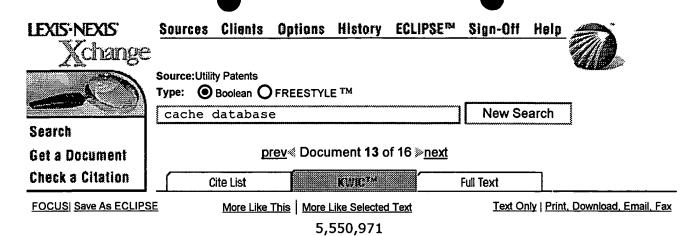
   CORE TERMS: bit, storage, secondary, directory, cache, processor, disk, database, updated, cached...

15. <u>5,408,653</u>, Apr. 18, 1995, Efficient data base access using a shared electronic store in a multi-system environment with shared disks, Josten, Jeffrey W., Morgan Hill, California Masatani, Tina L., San Jose, California Mohan, Chandrasekaran, San Jose, California Narang, Inderpal S., Saratoga, California Teng, James Z., San Jose, California, International Business Machines Corporation, Armonk, New York (02)

CORE TERMS: database, cached, buffer, designated, updating, store-dependent, update, log, storage, lock...

16. <u>4,876,731</u>, Oct. 24, 1989, Neural network model in pattern recognition using probabilistic contextual information, Loris, Keith, Brooklyn, New York Euchner, James, Bedford, New York, Nynex Corporation, New York, New York (02)

CORE TERMS: symbol, network, cell, unknown, selector, path, gnostic, contextual, neural, probability...

FOCUS| Save As ECLIPSE   Show Hits   Text Only | Print, Download, Email, Fax

Cite List	KWIC™	Full Text

prev◄ Docs **11 - 16 of 16** ► next

Source:	Utility Patents
Search Terms:	cache database
View:	CITE
Date/Time:	Saturday, June 19, 1999 - 8:34 PM EDT

Search | Get a Document | Check a Citation
Sources | Clients | Options | History | ECLIPSE | Feedback | Sign-Off | Help
About LEXIS-NEXIS | Terms and Conditions

LEXIS·NEXIS
Xchange

Sources　Clients　Options　History　ECLIPSE™　Sign-Off　Help

Source: Utility Patents
Type: ● Boolean ○ FREESTYLE ™

| cache database | New Search |

Search

Get a Document

Check a Citation

prev◄ Document **13** of 16 ►next

| Cite List | KWIC™ | Full Text |

FOCUS| Save As ECLIPSE　　　　　More Like This | More Like Selected Text　　　　Text Only | Print, Download, Email, Fax

5,550,971

GET 1st DRAWING SHEET OF 6

Aug. 27, 1996

Method and system for generating a user interface adaptable to various database management systems

INVENTOR: Brunner, Hans, Denver, Colorado
McCandless, Timothy P., Boulder, Colorado
Sparks, Randall B., Louisville, Colorado
Cuthbertson, Robert J., Boulder, Colorado
Durand, Jacques, Louisville, Colorado
Fogel, Steven M., Denver, Colorado

CORE TERMS: database, dot, user, interface, fot, query, entity, layer, meta-model, char...

DETDESC:

... updated in order to reflect the change in the user interface. All these changes can be made at run time without recompiling.

The local/networked computer system 14 further includes a local **cache database** 26. This local **cache database** is used for two purposes. First, the local **cache database** is used to store the search results of queries received from the remote database 12. If the query constraints that produced the search results are narrowed, it is more efficient to only search the data stored in the local **cache database** 26 rather than performing a new search on the remote database 12. The second purpose of the local **cache database** is to implement data layer of the semantic data model 24 that is used to instruct the database management system 10 how to generate the user interface. In the present embodiment of the invention, the local **cache database** 26 is implemented using a commercially available, object-oriented knowledge representation system ROCK TM manufactured by Carnegie Group, Inc. Finally, the local/networked computer system 14 includes an ...

... by the results of queries completed when the user queries the remote database. The results of a query are returned as lists to the query language processor 22 shown in FIG. 1 and stored in the local **cache database** 26. Once the local **cache database** is populated, any further searching that involves narrowing of the queries does not involve searching the remote database directly, but instead can be limited to searching the local **cache database.**

The local **cache database** is populated in a step 108 by first creating an instance of a DO type for each object returned by the remote database query. The attribute names and values are stored as data members of the new DO. Then, a pointer to the new DO is added to the

instanceset of the DOT in the model layer corresponding to the type of the new DO added.

After the data objects have been created in the local **cache database,** the relationships between the objects are added in a step 110. This is accomplished by creating instances of the classes defined in the model layer of the data model. First a new instance of the FO class is ...

... in the database that are related to a client entry having the name "U S WEST".

As described above, the results of queries to the database are used to populate the data layer of the database model that is stored in the local **cache database** in the manner shown at steps 108 and 110 of FIG. 3. Any further searching that narrows the query criteria is performed on the local **cache database.**

FIG. 5 is a flow chart showing the steps performed by the present invention to generate a graph of the model layer of the data model. In order to generate a corresponding graph of the model ...

... constraints at step 258. It is then determined if the software is operating in a "local" mode at step 262. If the system is operating in a local mode, the data to be searched have been previously stored in the local **cache database.** If not, the remote database is searched. Assuming the System is not operating in a local mode, the query constraints are translated into the remote database's own query language at a step 262. The remote database is then searched at a step 264 and the results are loaded into the local **cache database** at a step 266 as described above.

If the system is operating in a local mode and the data to be searched are already located in the local **cache database,** the local **cache database** is searched for all instances of data meeting the query constraints. In either case, the search results are displayed to the user at a step 270 in any of a number of ways which will be apparent to those skilled ...
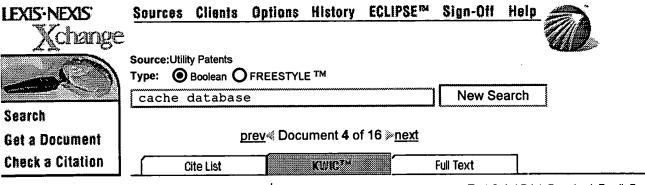
... returns Patent. The argument and the result belong the same layer among ML2, ML3 or ML4.

The following C + + listing shows how the data model is actually queried to generate the user interface or to search the local **cache database** that has been populated with objects. The function "ApplyFunction" is applied to an object (either of type DOT or type DO). The function takes as its argument the name of a FOT type that is connected to the DOT or DO object. The function returns the instances of the domain and ...

... Similarly if the function is applied to an instance of the DO class such as U S WEST arrow right ApplyFunction (FILES-PAT) would return all the patent applications filed by U S WEST that are stored in the local **cache database.** The function allows the FOT objects to be used as functions to query the data model itself or the local **cache database.** For example, applying the FOT "IN LITIGATION" to the DO "U S WEST" returns all instances of type LITIGATION that are either in the range or the domain of the type of the U S WEST object.

As can be ...

| Cite List | KWIC™ | Full Text |

LEXIS·NEXIS
Xchange

Source: Utility Patents
Type: ⦿ Boolean ◯ FREESTYLE ™

| cache database | | New Search |

Search

Get a Document

Check a Citation

| Cite List | KWIC™ | Full Text |

FOCUS| Save As ECLIPSE   More Like This | More Like Selected Text   Text Only | Print, Download, Email, Fax

*Pat. No. 5835908, ***

5,835,908

GET 1st DRAWING SHEET OF 15

Nov. 10, 1998

Processing multiple database transactions in the same process to reduce process overhead and redundant retrieval from database servers

INVENTOR: Bennett, John G., Bellevue, Washington
Dalal, Ketan, Seattle, Washington

CORE TERMS: database, cache, database-accessing, stream, lock, retrieved, computer system, identifier, server, caches...

ABST:
   The present invention uses a segmented caching data structure to **cache database** objects provided by a database server. The database server provides database objects in response to requests by a number of different programs. The segmented caching data structure is made up of ...

SUM:

... retrieved from a database including tables, fields, files, programmatic objects, and other units of data.

The facility executes multiple database-accessing programs in the same process, and utilizes a hierarchy of caches to **cache database** objects retrieved from a database server. The hierarchy of caches includes one program cache for each database-accessing program, and a single process cache. The facility uses each program cache to store database objects recently retrieved by the **cache's database**-accessing program, and uses the process cache to store database objects recently retrieved by any of the database-accessing programs. The program caches each allow a database-accessing program to quickly obtain a ...

DETDESC:

... programs needs a database object, the facility first searches the program cache for the database-accessing program and, if the program cache contains the database objects, the database-accessing program accesses it in its program **cache. If the database**-accessing program's program cache does not contain the needed database object, the database-accessing facility searches the process cache for the needed object, and, if the process cache contains the

needed database object, copies it to the program **cache where the database**-accessing program can access it and make changes to it without affecting the version of the database object used by other database-accessing programs in the same process. If the needed database object is in neither the ...

... space 245 in the memory of the database-accessing computer system 230. The data 150 includes a single process cache 281 having a lock 282. The facility uses the process cache 281 to **cache database** objects retrieved by any of the database-accessing programs executing in the process having process address space 245. The data 250 further includes a program cache for each data accessing program ...

... accessing programs in any number of processes. Also, the database server programs may execute on a data-accessing computer system instead of a dedicated database server computer system. Further, the facility may **cache database** objects obtained from more than one database server, either in a single combined cache or in separate, per-server caches.

FOCUS| Save As ECLIPSE   More Like This | More Like Selected Text   Text Only | Print, Download, Email, Fax

Cite List	KWIC™	Full Text

prev◄ Document **4** of 16 ►next

Source: Utility Patents
Search Terms: cache database
View: KWIC
Date/Time: Saturday, June 19, 1999 - 8:39 PM EDT

Search | Get a Document | Check a Citation
Sources | Clients | Options | History | ECLIPSE | Feedback | Sign-Off | Help
About LEXIS-NEXIS | Terms and Conditions

LEXIS·NEXIS
Xchange

Sources Clients Options History ECLIPSE™ Sign-Off Help

Search

Get a Document

Check a Citation

Source: Utility Patents
Type: ● Boolean ○ FREESTYLE ™

| cache database | | **New Search** |

prev◄ Document **11** of **16** ►next

| Cite List | KWIC™ | Full Text |

FOCUS| Save As ECLIPSE   More Like This | More Like Selected Text   Text Only | Print, Download, Email, Fax

Pat. No. 5682527, *

5,682,527

GET 1st DRAWING SHEET OF 19

Oct. 28, 1997

Method and apparatus for block-level auditing and database recovery in a transaction processing system

INVENTOR: Cooper, Thomas P., New Brighton, Minnesota
Hill, Michael J., Vadnais Heights, Minnesota
Konrad, Dennis R., Welch, Minnesota
Nowatzki, Thomas L., Shoreview, Minnesota

CORE TERMS: audit, database, cache, outboard, processing, data processing, segment, disk, storage, packet...

SUM:

... a system or disk failure by reading audit records that contain only changed blocks and copying the changed blocks to appropriate segments of the database without incurring increased recovery time.

Another object is to **cache the database** in an outboard file cache and recover the database using the outboard file cache.

Yet another object is to automatically determine whether the outboard file cache is available and use the outboard file cache for recovery if it is ...

FOCUS| Save As ECLIPSE   More Like This | More Like Selected Text   Text Only | Print, Download, Email, Fax

| Cite List | KWIC™ | Full Text |

prev◄ Document **11** of **16** ►next

Source:	Utility Patents
Search Terms:	cache database
View:	KWIC
Date/Time:	Saturday, June 19, 1999 - 8:35 PM EDT